Richard S. Sutton and Andrew G. Barto (1998)
Reinforcement Learning: An Introduction

Chapter 6:

Temporal-Difference Learning

Paul Wagner

T-61.6020 Reinforcement Learning - Theory and Applications
Spring 2006

Outline

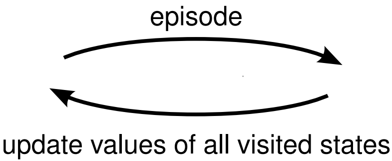
Outline

General Properties

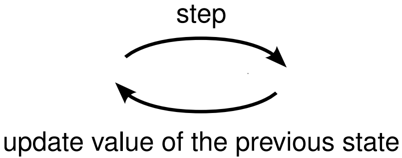
- In contrast to Dynamic Programming, both Temporal-Difference (TD) and Monte Carlo (MC) methods are model-free
- The environment is sampled by actually executing the current policy
- Temporal-Difference and Monte Carlo methods differ in the way they update the state value estimates

Updating Of The Value Estimates

- MC: Only the true observed total reward is used for value updates:

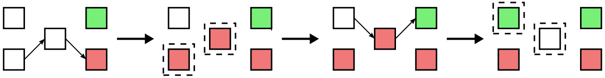


- TD: Also values of all intermediate states are used: (estimating from estimates: *bootstrapping*)

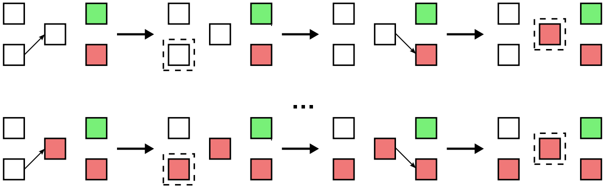


An Example

- Monte Carlo



- Temporal-Difference



Value Estimate Update Rules

- Monte Carlo (every-visit):

$$V(s_t) \leftarrow V(s_t) + \alpha[R_t - V(s_t)]$$

  - Update target R_t: the true observed total reward following time t
  - Assigns the true observed total reward directly to all participated states

- Temporal-Difference (TD(0)):

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

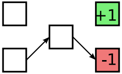
  - Update target $r_{t+1} + \gamma V(s_{t+1})$: immediate reward plus the *estimated* value of the next state (discounted by γ)
  - Keeps the value estimates consistent (true rewards propagate backwards on state transitions)

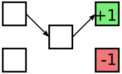
Speeding Up Temporal-Difference Learning: TD(λ)

- Multiple states can be updated on every step
- Keep a record of recently visited states: eligibility traces
- Exponential decay of the traces: decay factor λ
- On every step, update all states marked by the traces: observed rewards propagate faster
- TD(λ): Temporal-Difference Learning With Eligibility Traces

Different Optimalities: Batch Training With Limited Experience

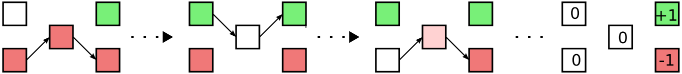
- Methods converge to different results
- Only two episodes available:

E1: 

E2: 

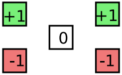
- Monte Carlo:

- Temporal-Difference:



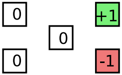
Different Optimalities: The Result

- Monte Carlo



  - Optimal solution for the data set
  - Structure of the right side “leaked” to the left: the Markov property is ignored

- Temporal-Difference



  - Sub-optimal solution for the data set, but the Markov property is respected
  - Optimal solution assuming an underlying Markov model with structure suggested by the data
  - Converges to the *certainty-equivalence estimate*

Advantages of Temporal-Difference Methods

- Do not require a model
  - learn directly from the environment
- On-line operation
  - no need to wait until the end of an episode
  - very long episodes
  - continuing tasks with no episodes at all
- Use of the Markov property: more data-efficient learning

Outline

Model-Based and Model-Free Methods

- Model-based
  - Dynamic Programming (DP)
  - full backups: current policy not actually executed
  - State values $V(s)$ are sufficient
  - Current policy does not affect which parts of the state space are sampled: all states are swept through systematically
- Model-free
  - Monte Carlo (MC), Temporal-Difference (TD)
  - sample backups: the environment is sampled by following the current policy
  - Actions are actually taken: action values must be separated from state values: action-value function $Q(s, a)$
  - Current policy *does affect* which parts of the state space are sampled (and which are not!): exploration sensitivity

Exploration-Exploitation Tradeoff

- The current policy determines which parts of the state space are sampled
- A greedy policy might converge to a sub-optimal solution and get stuck in it
- The policy must exhibit *exploration*: it has to keep selecting also sub-optimal actions $\rightarrow$ a *soft* policy
- But sub-optimal actions lead to sub-optimal rewards in short-term future
- How should exploration of the environment and exploitation of the current knowledge balanced?

Learning The Value Function Under Exploration

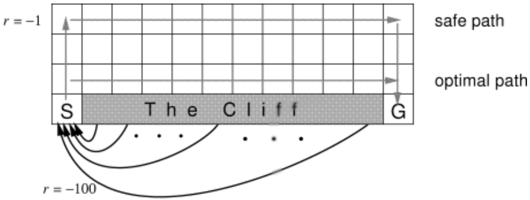
- Some future policy has to be assumed when computing a value estimate
- Should the value estimates be computed for the exploring, sub-optimal policy or directly for a greedy one?
- On-policy vs. off-policy

On-Policy and Off-Policy

- Behavior policy π' and estimation policy π

	DP (model-based)	MC and TD (Model-free)	
		on-policy	off-policy
behavior π' estimation π	$\pi' = \pi =$ *greedy* (π' irrelevant)	$\pi' = \pi =$ *soft* $Q^\pi \nrightarrow Q^*$	$\pi' =$ *soft* $\pi =$ *greedy*

- Safe and optimal path under exploring policy



TD Control With On-Policy and Off-Policy Methods

- On-Policy TD Control: Sarsa
  - State values evaluated for the behaviour policy
  - The soft behaviour policy must be adjusted towards a greedy policy over time to allow convergence

- Off-Policy TD Control: Q-Learning
  - State values are always evaluated for a greedy policy
  - The action-value function Q directly approximates Q^*, the optimal action-value function, independent of the policy being followed

Q-Learning

- Off-policy Temporal Difference control: model-free, on-line, exploration insensitive, easy to implement
- One of the most important breakthroughs in reinforcement learning

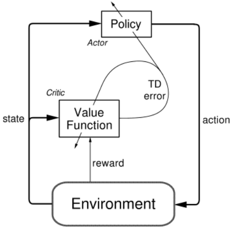
- One-step Q-learning:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

  - Update target $r_{t+1} + \gamma \max_a Q(s_{t+1}, a)$: immediate reward plus the estimated value of the *best available* next action (discounted by γ)
  - Q is directly computed for the greedy policy

Outline

Actor-Critic Methods



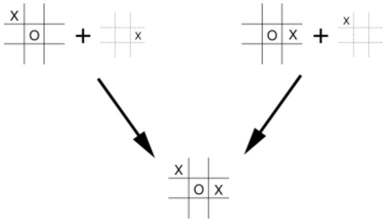
- Explicitly stored policy separate from the value function
- Minimal computation in order to select actions
- Difficult to get the relative learning rates of the components right
- Can learn an explicitly stochastic policy: competitive and non-Markov cases
- More appealing as psychological and biological models

Undiscounted Continuing Tasks: R-Learning

- Undiscounted continuing case: an advanced version of the reinforcement learning problem
- R-learning:
  - Estimate the average expected reward per time step
  - Starting from some states, the short-term expected reward is better
  - Try to stay in these areas
  - Off-policy TD: is a variation of Q-learning
- The method is considered experimental

Games and Afterstates

- In games, the immediate result of an action is usually known: *an afterstate*
- Multiple transitions might lead to the same afterstate:



- Break the environment's dynamics into the immediate effect and the unknown random process
- Don't estimate current positions and possible moves, estimate the resulting afterstates

Outline

Summary

- Temporal-Difference learning
  - Model-free, on-line
  - Step-by-step value updating, make the estimates consistent
  - Markov property respected: converges to the certainty-equivalence estimate
- Q-learning:
  - off-policy TD control
  - exploration insensitive

- Actor-Critic methods: explicit policy
- R-learning: undiscounted continuing tasks
- Games and afterstates